Apricot
Modem

Modem
Driver

Modem Driver

Contents

Introduction

The Modem driver software (VR2.0) is a loadable device driver which runs under MS-DOS 2.11 in conjunction with the generic Apricot ROM BIOS (or the RAM based BIOS release VR2.6 and later on the Apricot pc/xi range of computers). It is supplied with the standard release software with all the generic ROM BIOS based Apricots and also with the pc/xi machines incorporating BIOS release VR2.6 and derivatives.

The driver forms the interface between applications software and the hardware of the optional ACT multi-mode, auto dial, auto answer, integral Modem.

The Modem driver supports all the features detailed below.

1. Auto dial - data call connection
2. Auto dial - voice call connection
3. Auto answer
4. CCITT V21 300 baud Full Duplex
5. CCITT V23 1200/75 baud split speed Full Duplex
6. CCITT V23 1200 baud Half Duplex
7. Selectable parity generation and checking
8. XON/XOFF flow control

This allows applications software to configure the Apricot to act as a multi-purpose communicating microcomputer with a vast and diverse range of differing capabilities.

A few examples of possible uses the Apricot-integral Modem combination can be employed to do are detailed below:

1. Interface to the majority of public/private data base services currently available via the PSTN (e.g. Prestel and all the other protocol compatible services, Telecom Gold, etc).
2. Emulate various standard computer terminals used for communicating to mainframes and minicomputers.
3. Provide repertory dialler facilities for voice call connection.
4. General purpose networking for transferring files and data to other microcomputers linked together via the PSTN.

```
810 REM ***************************************
820 REM * INITIALISE MODEM STATUS ARRAY
830 RESTORE 880
840 FOR STATUS%=0 TO 31
850 READ STATUS$(STATUS%)
860 NEXT
870 RETURN
880 DATA "Idle"
890 DATA "Looking for dial tone"
900 DATA "Dialling"
910 DATA "Looking for answerback"
920 DATA "Voice connection made"
930 DATA "Waiting for DCD"
940 DATA "UNKNOWN"
950 DATA "UNKNOWN"
960 DATA "UNKNOWN"
970 DATA "Waiting for DTR"
980 DATA "Idle in answer mode"
990 DATA "UNKNOWN"
1000 DATA "UNKNOWN"
1010 DATA "UNKNOWN"
1020 DATA "UNKNOWN"
1030 DATA "UNKNOWN"
1040 DATA "UNKNOWN"
1050 DATA "Call abandoned (no dial tone)"
1060 DATA "UNKNOWN"
1070 DATA "Call abandoned (no answerback)"
1080 DATA "Voice call timeout"
1090 DATA "Call abandoned (no DCD)"
1100 DATA "UNKNOWN"
1110 DATA "Connected"
1120 DATA "UNKNOWN"
1130 DATA "Call abandoned (no DTR)"
1140 DATA "UNKNOWN"
1150 DATA "UNKNOWN"
1160 DATA "Ringing detected"
1170 DATA "UNKNOWN"
1180 DATA "Command accepted"
1190 DATA "Command invalid"
```

************* WARNING *************

The Modem with its associated Modem device driver software have been approved for connection to the British Telecom Public Switched Telephone Network. This approval is *invalidated* if the apparatus is subjected to any modification in any way not authorized by BABT or it is used with or connected to:

1. Modem device driver software that has not been formally accepted by BABT.
2. Any applications or any other external control software which causes the operation of the Modem to contravene the requirements of the standards set out in BABT/SITS/82/002S, 83/003S/RI and 82/005S.

Applications programmers should take particular attention to the following points.

1. Applications software must not attempt to access the Modem hardware directly. All requests and commands to the Modem must be made through the Modem device driver.
2. On no account must any attempt be made to modify the Modem device driver software.

To ensure that it is not possible for third party software to contravene BABT regulations, details of how the driver communicates with the hardware of the Modem and the actual device driver program listings are not and never will be made available.

All apparatus connected directly or indirectly to the British Telecom Public Switched Telephone Network must be approved apparatus as defined in Section 16 of the British Telecommunications Act 1981.

****

```
380 REM ************ POLL TRANSMIT AND RECEIVE ************
390 IF (R%(2) AND &H100) = 0 THEN 450        :REM CHARACTER RECEIVED ?
400 IF (R%(2) AND &H8000) = 0 THEN 430       :REM RECEIVE ERROR ?
410 STATUS%=R%(2) :GOSUB 670                 :REM REPORT ERROR
420 IF STAT% > 7 THEN 170                     :REM TERMINATE IF FATAL
430 PRINT RXC$;                               :REM PRINT RECEIVED DATA
440 GOSUB 580                                 :REM SET UP A NEW RECEIVE JOB
450 IF (T%(2) AND &H100) = 0 THEN 510        :REM TRANSMIT DONE ?
460 Q$=INKEY$ :IF Q$="" THEN 390             :REM KEY PRESSED ?
470 IF (T%(2) AND &H8000) = 0 THEN 500       :REM TRANSMIT ERROR ?
480 STATUS%=T%(2) :GOSUB 670                 :REM REPORT ERROR
490 GOTO 170                                  :REM FATAL ERROR TERMINATE
500 GOSUB 520                                 :REM SET UP NEW TRANSMIT JOB
510 GOTO 380                                  :REM POLL FOREVER
520 REM ************************************************
530 REM SET UP A NEW TRANSMIT JOB
540 LSET TXC$=Q$
550 T%(1)=41 :T%(2)=0 :T%(3)=1 :T%(5)=0
560 GOSUB 140
570 RETURN
580 REM ************************************************
590 REM SET UP A NEW RECEIVE JOB
600 R%(1)=40 :R%(2)=0 :R%(3)=1 :R%(5)=0
610 GOSUB 120
620 REM ************ POLL MODEM STATUS AND REPORT ************
630 S%(1)=20 :S%(2)=0 :S%(3)=0 :S%(4)=0 :S%(5)=0
640 GOSUB 130
650 STATUS%=S%(2) :GOSUB 670
660 RETURN
670 REM ************************************************
680 STAT%=(STATUS% AND 255)
690 PRINT CHR$(27);"J";CHR$(27);"Y";CHR$(32+24);CHR$(32);
700 PRINT (STATUS% AND 255);" - ";
710 IF (STAT% < 64) AND (STAT% > 31) THEN PRINT STATUS$(STAT%-32);
                            ELSE PRINT CHR$(7);
720 IF STAT% = 255 THEN PRINT "Timeout";
730 IF STAT% = 125 THEN PRINT "Invalid number";
740 IF STAT% = 126 THEN PRINT "No re-dialing allowed";
750 IF STAT% = 127 THEN PRINT "Call list full";
760 IF (STATUS% AND &H200) <> 0 THEN PRINT " BUSY";
770 IF (STATUS% AND &H8000) <> 0 THEN PRINT " ERROR";
780 PRINT CHR$(27);"K";
790 PRINT CHR$(27);"k";
800 RETURN
```

Applications Interest

Installation

The loadable device driver is contained within a file named MODEMAPR.SYS. The process to install the Modem device driver is the standard procedure required to install all new loadable device drivers.

All that is necessary, is to copy the Modem driver file onto a system disk and then edit the CONFIG.SYS file using EDLIN or WORDSTAR. The following line must be appended to the end of CONFIG.SYS which will then cause the driver to be loaded upon the next system reset.

```
DEVICE=MODEMAPR.SYS
```

Applications Interface

The application interacts with the driver through the standard MS-DOS requests for file I/O, although the meaning of the parameters passed in this way is not the same as with standard file I/O. This enables applications to be written in any programming language which supports READ and WRITE to standard devices.

The interface requires the application to set up a request block and then write its address to the file *MODEM*. The format of the request block is defined below.

Request block format

Command	(word)
Status	(word)
Byte Count	(word)
Address	(doubleword) or **Data** - Command dependent

In some cases the address/data and byte count words are redundant. No further information has to be passed to or from the driver as all information required by the Modem is inherent in the command itself.

Example Application in Basic - Teletype Simulation

```
10 0FF=2                                        :REM 1 FOR INTERPRETED, 2 FOR COMPILED
20 WIDTH 255                                    :REM DISABLE BASICS 'CR','LF'S
30 PRINT CHR$(27);"x1";                         :REM ENABLE 25TH STATUS LINE
40 DIM STATUS$(50)                              :REM INITIALISE STATUS MESSAGES
50 GOSUB 810
60 DIM D%(5),R%(5),T%(5),S%(5)
70 OPEN "R",#2,"DUMMY"                          :REM MODEM REQUEST HEADERS
80 FIELD #2,1 AS RXC$,1 AS TXC$,20 AS DIAL$     :REM FIX A FEW VARIABLES IN MEMORY
90 GOTO 170
100 REM *****************************************************************
110 A$=MKI$(VARPTR(D%(1)))+MKI$(O): GOTO 150
120 A$=MKI$(VARPTR(R%(1)))+MKI$(O): GOTO 150
130 A$=MKI$(VARPTR(S%(1)))+MKI$(O): GOTO 150
140 A$=MKI$(VARPTR(T%(1)))+MKI$(O)
150 OPEN "O",#1,"MODEM" : PRINT #1,A$;: CLOSE #1
160 RETURN
170 REM ************************************ DIAL CALL ******************
180 PRINT "PHONE NUMBER - ";:INPUT NUMBER$
190 D%(1)=30 :D%(2)=0 :D%(3)=0 :D%(4)=0 :D%(5)=0
200 D%(3)=LEN(NUMBER$)
210 LSET DIAL$=NUMBER$
220 POKE VARPTR(D%(4)),PEEK(VARPTR(DIAL$)+OFF)
230 POKE VARPTR(D%(4)+1,PEEK(VARPTR(DIAL$)+OFF+1)
240 GOSUB 110
250 GOSUB 620
260 IF (D%(2) AND &H100) = 0 THEN 250           :REM DISPLAY CALL STATUS
270 STATUS% = D%(2) :GOSUB 670                  :REM WAIT FOR COMPLETION
280 IF (STATUS% AND &H8000) <> 0 THEN 170       :REM DISPLAY COMPLETION STATUS
290 REM ******************** TELETYPE MODE ********************          :REM CALL FAILED ?
300 PRINT "TELETYPE MODE"
310 REM A$=""" :D$=""
320 POKE VARPTR(R%(4)),PEEK(VARPTR(RXC$)+OFF)
330 POKE VARPTR(R%(4)+1,PEEK(VARPTR(RXC$)+OFF+1)
340 POKE VARPTR(T%(4)),PEEK(VARPTR(TXC$)+OFF)
350 POKE VARPTR(T%(4)+1,PEEK(VARPTR(TXC$)+OFF+1)
360 T%(2)=&H100                                 :REM INITIAL TX JOB DONE FLAG
370 GOSUB 580                                   :REM SET UP INITIAL RECEIVE JOB
```

The simple example below shows how to interface to the Modem from a BASIC program. A more complete application is listed at the end of this document.

```
100 FF=2                                    REM 1 FOR INTERPRETED, 2 FOR COMPILED
100 DIM M%(5)
110 OPEN "0",#1,"MODEM"
120 OPEN "R",#2,"DUMMY"                      REM just so we can field things
125 FIELD #1, 128 AS Z$                      REM data must not move
130 REM set up request block
140 M%(1)=1                                  REM command
150 M%(2)=0                                  REM returned status word
160 M%(3)=128                                REM byte count
170 REM set up pointer to BASIC string data
180 POKE VARPTR(M%(4)),PEEK(VARPTR(DIAL$)+0FF)
190 POKE VARPTR(M%(4))+1,PEEK(VARPTR(DIAL$)+OFF+1)
200 M%(5)=0                                  REM BASIC wont tell you the segment
210 A$=MKI$(VARPTR(T%(1)))+MKI$(0)
220 OPEN "0",#1,"MODEM" : PRINT #1,A$::CLOSE #1
230 PRINT #1,M%(1)
240 REM wait for done bit to be set
250 IF ((M%(2) AND &H10)=&H10) THEN 250
260 REM process data
```

The application program sets up its parameter block in memory and writes its' address to the *MODEM* device. The driver then returns control to the user, whose responsibility it is to monitor the status word in the parameter block until the requested transfer is complete.

Note: When writing applications it is essential to ensure that once the addresses of the paramater block and other data have been set up that they do not change until the request is completed. Failure to do this will result in unpredictable system failures. The BASIC language in particular has a nasty habit of moving data around, especially strings. TAKE CARE !!!

The next part of this chapter details the format and meaning of the commands which can be sent to the driver. This is then followed by a section detailing all the possible status responses returned by the driver after issuing a command.

Ringer Equivalence

Equipment for attachment to the PSTN is assessed to determine its' Ringer Equivalence Number (REN). The REN indicates, in effect, the load that the telephone exchange sees when ringing the equipment. It is not permitted to put more than a total of 4 REN onto any exchange line. The ADM/2 has a REN of 3 and care must be taken not to use it with other telephone equipment that would result in the maximum figure of 4 REN being exceeded.

To determine the maximum number of items of apparatus that can be connected simultaneously to an exclusive line, the total REN is obtained by summing the REN values of each item connected to the line. All BT supplied instruments should be assumed to have a REN value of 1.0 unless otherwise marked.

Mark of Origin

The ADM/2 Modem is manufactured in the UK and supplied by ACT (Computers) Ltd.

Modem Function

The ADM/2 Modem is a frequency shift keyed (FSK) Modem conforming to CCITT V21 (300 bps full duplex) and CCITT V23 (1200/75 bps full duplex) standards. It has auto answer capability conforming to CCITT V25 and also incorporates an integral loop disconnect (pulse) autodialler.

Commands

All commands may be executed in two modes, unless otherwise stated.

Mode-1

If the command is issued as a positive number, the driver will initiate the command, and return control to the user immediately. It is then the users responsibility to monitor the status word in the command packet and wait for completion.

Mode 2

For operation in a simple program the commands can be negated, in which case control will not be returned to the user until the command has been completed. The status word in the command packet will then have been set according to, whether the command was succesful or not.

The commands are subdivided into a number categories in the following pages. These are as follows:

1. Initialisation Commands.
2. Setting up Commands.
3. Request for Status.
4. Auto dial Commands.
5. Data transfer Commands.
6. Auto answer Commands.

Applications Documentation

Another requirement of BABT is that the information detailed below must be included with the user instructions. As the user instructions will be supplied with the application, all software documentation instructing the user how to interface to a particular application must incorporate this information in the body of the document.

Statutory Information

BABT Approval No.

S/1397/3/E/500039

Model Number

The Modem referred to in this chapter is the ACT Apricot integral multi-standard autodialling Modem, ADM/2.

BT Circuits

The Modem is designed to be used on 2-wire PSTN circuits only. It generates a CCITT V25 answer sequence when set in auto answer mode and may be used on lines listed in British Telecom telephone directories. It must not be used with payphones, party lines, private lines maintained by British Telecom or with certain types of call connect systems that do not use two wire signalling systems.

Bell Tinkle

When the Modem is used with telephones that use a mechanical bell, "bell tinkle" will be caused when dialling.

1. Initialisation Commands - Commands 1 to 3

Command 1 - Initialise driver

This command re-initialises the Modem and selects the default parameters. It is executed automatically when the device driver is loaded. All data buffers are cleared and any calls in progress are aborted, leaving the Modem in idle mode (waiting for DTR).

Command   1
Status   —
Byte Count —
Address   —

The Initialise Driver Command sets the driver to the default status as listed below. Their values can be changed by issuing the appropriate Setting Up Command as described in the next few pages. For a detailed discussion on the meaning of each default state, reference should be made to the appropriate command.

Default values

Command	Default State
4	40 second timeout
5	wait for dial tone
7	CCITT V21 300/300 baud full duplex
8	send
9	no timeout
10	no timeout
11	even parity/no flow control for both transmit and receive
12	Auto turn round using the ASCII EOT character

The second BABT requirement is taken care of by using the failed call list. This is a 256 byte list resident within the driver memory space. It is used to store any number which was used for a dial attempt and the result of the attempt was a failed connection (e.g. if the number was engaged).

The first operation the driver does on receiving a Dial Command is to check if the failed call list is full. If it is, the command is aborted and error status is generated. The only method of clearing the failed call list in this situation is to manually reset the machine using the system reset button.

If there is space within the failed call list, the driver checks to the list to see whether an unsuccessful dial attempt has already been made to the number. If no such number exists in the list, the number is dialled. If the number is there, the driver checks for two other conditions, whether:

1. 4 attempts to dial the number have already been made without success.
2. Dialling the same number has been tried within the last minute.

If condition 1 is true, the command is aborted and error status is generated. The only method to continue using the Modem in this situation is to manually reset the machine using the system reset button.

If condition 2 is true, the command is aborted and error status is generated to indicate that a minute has not elapsed since the last attempt. The failed call list is not updated as no attempt is made to call the number. Successful connection automatically clears the number from the list. Unsuccessful connection causes the failed call flag to be updated with the number of unsuccessful attempts and a one minute background timer to be initiated.

Command 2 - Flush buffers

Clears the Modem receive buffer of all data. This is also automatically done whenever a new call is connected. It does not affect the status of the Modem or the contents of the failed call list.

Note: It is recommended that when a call is connected the application should wait a few seconds and then issue a flush buffers command to discard any invalid data. The precise time to wait is dependent on the particular service the Modem is being used for.

Command   2
Status   —
Byte Count —
Address   —

Command 3 - Abort Call

Instructs the Modem to terminate any call in progress and returns the Modem to idle status.

Command   3
Status   —
Byte Count —
Address   —

This command should be issued when a call is terminated by the remote modem. The purpose of this is to clear any error status flags and return the modem to the idle state.

Systems Interest

The Modem device driver handles all communication between the application and the Modem hardware, and protects the telephone network from misuse by preventing invalid requests being sent to the Modem from within applications programs. The driver has been formally accepted by BABT for use with the Modem. This is a pre-requisite for connecting the Modem to the PSTN run by British Telecom.

The only BABT regulation that is of particular interest to the applications programmer is the one concerning repeat auto dialling. This also explains the use of the failed call list which is mentioned several times within the Applications Interest section.

BABT Regulations for Autodialling

In order to conform with BABT regulations the following restrictions are placed on autodialling numbers.

1. No dialling is allowed to the emergency services.
2. The maximum number of times a number can be dialled by autocalling apparatus without making a connection is 4 with a minimum of a one minute delay between each attempt. Attempts to dial the same number can be repeated only if there is manual intervention by the USER.

The Modem device driver takes care of the first requirement by declaring any number within a Dial Command with the format as detailed below invalid, terminating the command operation and returning error status to the application.

Invalid Format:

999Y and **X999Y**

where **X** may be any single digit used to get an external line on a private exchange and **Y** represents any or no trailing digits.

Status Condition

01	Parity Error (receive data) **
02	Framing Error (receive data) **
04	Overrun Error (receive data) **
08	Break condition (receive data) **
20	Idle following reset
21	Looking for dial tone
22	Dialling
23	Looking for answerback
24	Voice connection made
25	Looking for carrier
29	Idle - Waiting for DTR
2A	Idle - Answer mode
31	Abandoned - No dial tone
33	Abandoned - No answerback
34	Voice call timeout
35	Abandoned - No carrier
37	Connected
38	Disconnected
39	Abandoned - No DTR
3C	Ringing detected
7C	Waiting 1 minute before re-dial
7D	Invalid phone number
7E	Number not allowed (4th re-dial)
7F	Call list full
80	Data invalid
81	Command invalid
FD	Line turn round failed
FE	Receive timeout elapsed
FF	Transmit timeout elapsed

** Multiple receive errors conditions can occur. e.g. error code 03 hex signifies Parity and Framing error.

2. Setting up Commands - Commands 4 to 12

Default status set by the Initialise Driver command (Command 1) is marked by an *.

Command 4 - Set Timeout on dialling

Sets the timeout period after which a dial attempt will be abandoned if the call has not been connected. The Modem must be idle when this command is invoked otherwise a BUSY error status will be returned.

Command	4
Status	—
Byte Count	—
Data	Valid data

Valid data (Dec. Value)

10	-	10 sec timeout
20	-	20 sec timeout
30	-	30 sec timeout
40	-	40 sec timeout *

Command 5 - Set Dial Tone Detect

Selects either a short delay before automatically dialling the number or instructs the Modem to wait until the dial tone is detected before dialling. In both cases, the time the Modem is prepared to wait before the call is connected is determined by the data of command 4. The Modem must be idle when this command is invoked otherwise a BUSY error will result.

Command	5
Status	—
Byte Count	—
Data	Valid data

Valid data

0	-	Wait for dial tone before dialling *
1	-	4 second delay before dialling

Command 6

Not implemented.

Command 7 - Set Mode

Selects the current mode of operation. The Modem must be idle when this command is invoked otherwise a BUSY error will result.

Command   7
Status   —
Byte Count   —
Data   Valid data

Valid data

0 - CCITT V23 1200/75 baud, Full duplex
2 - CCITT V21 300/300 baud, Full duplex *
3 - CCITT V23 1200/1200 baud, Half duplex

Command 8 - Set Channel

This command is only effective in the half duplex mode. If the send channel is selected, the Modem is set to be the *calling* station and can transmit data at 1200 baud. If the receive channel is selected, the Modem is set to be the *answering* station and is able to receive 1200 baud data.

Note: Both *calling* and *answering* stations must define the protocol to agree about this.

Command   8
Status   —
Byte Count   —
Data   Valid data

Valid data

0 - send *
1 - receive

Status Word

When commands are initiated and control returned to the application program, it is the applications responsibility to test the status word in the request packet for completion. In any case, the word should always be checked for error conditions. When the command is initiated, the status word is automatically set to zero so the application program need not do this.

The format of the status word is detailed below. The high byte contains flags which are used to signify completion of the command and also indicate the reason for command termination (i.e. successful completion or command aborted due to a busy or error condition). The lower byte is used to signify actual error conditions and also for routine status monitoring as a call set up sequence proceeds.

Status Word Format

bit	15	8	7	0
	Command Flags		Status Condition	

Command Flags

Bit 15   Error - set if an error occurs due to the command not being able to be carried through sucessfully. The cause of the error is specified in the Status Condition byte.

Bit 09   Busy - set if the command could not be carried out due to another conflicting command being already in progress.

Bit 08   Done - set when the command has been completely processed or has been terminated due to an error.

Note: Bits 10 to 14 are not used and are always set low.

Command 9 - Set Receive Timeout

Sets a timeout within the driver to prevent an infinite wait on received data. Each time a read request is made to the driver this timeout is initiated.

Command	9
Status	—
Byte Count	—
Data	Valid data

Valid data

0 to 65,535   The timeout in seconds is calculated by dividing the data by 10 (e.g. Max possible value is 65,535/10 seconds).

* default is 0, disables the timeout.

Command 10 - Set Transmit Timeout

As above for Command 9 but for the transmit channel. Each time a transmit request is made to the driver, the timeout is invoked. The format of the valid data for the timeout is identical to Command 9 data.

Command	10
Status	—
Byte Count	—
Data	Valid data

* default is 0, disables the timeout.

Command 11 - Select Protocol

Selects the parity checking/generating and flow control methods for the transmit data and for what is expected in the receive data. The data consists of four bytes, two for transmit data and two for receive data.

Command	11
Status	—
Byte Count	Length of protocol data (4)
Address	Start address of protocol data

Protocol data

Byte 0	-	Rx parity
Byte 1	-	Tx parity
Byte 2	-	Rx flow control
Byte 3	-	Tx flow control

Auto Answer Command - Command 50

Command 50 - Set Auto Answer Mode

Sets up the Modem board to answer and respond to an incoming call. Note that if a receive data command has not also been issued data will be lost.

Note: It is recomended that when a call is connected the application should wait a few seconds and then issue a flush buffers command to discard any invalid data.

Command	50
Status	—
Byte Count	—
Address	—

Parity selection (Byte 0/Byte 1) valid data

0 - no parity
1 - no parity
2 - even parity *
3 - odd parity

Flow control (Byte 2/Byte 3) valid data

0 - no flow control *
1 - XON-XOFF flow control

Command 12 - Turn Round Protocol

In half-duplex mode, selects whether the line is to be turned round automatically by a turn round character. The normal turn round character is the ASCII End of Transmission (EOT) - 04H.

Both ends of the line of the receive and transmit stations must be synchronised correctly to avoid loss of carrier. The transmitting station must extend the transmission after the EOT character by sending a few null data bytes (approximately 4) to allow the receiving station time to accept the EOT character and turn it's end of the line round.

When the line has been turned round, a delay should be implemented to allow the line to settle before any data is transmitted.

Prior to using the half-duplex mode, the line should be set up so that one station is ready for transmitting and the other is ready for reception. This must be done using command 8, before the call is dialled or answer mode is selected.

The auto turn round protocol may be changed whilst the modem is on line to allow transparent data transfers. This allows the EOT character is treated as a normal charcter until auto line turn round is required again.

Command   12
Status   —
Byte Count 2
Address   Start address of turn round data

Valid Data

Byte 0 - 0 = Auto turn round disabled
           1 = Auto turn round enable *
Byte 1 - Turn round character EOT (Default 04H)

Command 43 - Terminate Tansmit

This command enables the application writer to terminate the transmission of data prior to normal completion of the transfer. Upon completion of the termination operation, the *Byte count* value will be the number of bytes transmitted from the applications buffer.

Command   43
Status   —
Byte Count —
Address   —

Command 44 - Send Break

This command enables the application writer to signify a break in the transmission of data. The break character is formed by a series of spaces. The duration of the break is selectable in increments of one tenth of a second.

Command   44
Status   —
Byte Count —
Data   Duration of Break

The duration of the break is determined by multiplying the value programmed int the data field by a tenth (e.g. for a break of a second, the data would be 0AH). The value of 00H is invalid.

5. Data Transfer Commands - Command 40 to 44

Command 40 - Receive Data

This command initiates reception of data by setting up a buffer pointer and a Byte Count. The Byte Count indicates space available (in bytes) in the applications receive data buffer (maximum of 64 Kbytes).

Upon completion of the transfer, the Byte count value is changed to indicate the number of bytes received.

If a receive error condition occurs, the operation is terminated and the Byte Count is set to indicate the number of characters received.

The last character written into the buffer will be the one containing the error, all previous characters should be regarded as valid.

Command	40
Status	—
Byte Count	Number of bytes received
Address	pointer to data buffer

Command 41 - Transmit Data

Initiates transmission of data by setting up a buffer pointer and a byte count. Upon completion the *Byte count* value is the number of bytes transmitted (maximum 64 Kbytes).

Command	41
Status	—
Byte Count	Number of bytes transmitted
Address	pointer to transmit buffer

Command 42 - Terminate Receive

This command enables the application writer to terminate the reception of data prior to normal completion of the transfer. Upon completion of the termination operation, the *Byte count* value will be the number of bytes received into the applications buffer.

Command	42
Status	—
Byte Count	—
Address	—

3. Status Requests - Command 20

Command 20 - Return Status

This command allows the user to check on the state of the Modem. Information is returned in the form of a status byte, within the Status word.

Command	20
Status	—
Byte Count	—
Address	—

Status returned

20	Idle following reset
21	Looking for dial tone
22	Dialling
23	Looking for answerback
24	Voice connection made
25	Looking for carrier
29	Idle - Waiting for DTR
2A	Idle - Answer mode
31	Abandoned - No dial tone
33	Abandoned - No answerback
34	Voice call timeout
35	Abandoned - No carrier
37	Connected
38	Disconnected
39	Abandoned - No DTR
3C	Ringing detected

Note: The idle conditions detailed above (codes 20, 29) should be treated identically, and mean that the modem is completely idle. The Answer mode idle condition (code 2A) has a different meaning and signifies that the modem is monitoring the line, waiting for a ringing tone to answer a call. This mode can be terminated by an abort call command which will reset the modem to a true idle condition.

4. Auto Dial Commands - Command 30 to 31

Command 30 - Dial (Data call)

This command checks for four conditions before dialling. These are whether:

1. The telephone number has been flagged four times in the failed call list (see Systems Interest for more details).
2. The failed call list is full.
3. The number is the number for dialling the emergency services (999).
4. An attempt has been made to ring the number within the last minute.

If all conditions are false, dialling is initiated. If any of the conditions are true, the call sequence is immediately aborted and error status returned.

If the dialling fails, the number is added to the failed call list (if not already there), or flagged again (if already present). Error status is also returned. Attempts to re-dial a number which has been unsuccessfully dialled four times previously or if the failed call list is full, results in the driver returning an error status requiring the *USER* to reset the system to flush out the failed call list.

The Modem must be idle when the dial command is invoked otherwise a BUSY error will result.

Note: It is recommended that when a call is connected the application should wait a few seconds and then issue a flush buffers command to discard any invalid data in the receive buffer.

The precise time to wait is dependent on the service being accessed.

Command	30
Status	—
Byte Count	length of number (maximum of 16 bytes)
Address	pointer to telephone number

The telephone number is a sequence of digits, each digit represented by the equivalent ASCII value (e.g. digit 7 is represented by 37 hex). ASCII B (42 hex) can also be incorporated anywhere within the sequence to introduce a 4 second delay.

Command 31 - Dial (Voice call)

This command functions in a similar way to command 30. Before initiating dialling, the command checks the failed call list and the other conditions as described above for data call connection. Dialling is initiated providing all the conditions are false. If any of the conditions are true the same sequence of events are initiated as described for data calls.

The Modem must be idle when this command is invoked otherwise a BUSY error will result.

Command	31
Status	—
Byte Count	length of number (maximum of 16 bytes)
Address	pointer to telephone number

The telephone number is a sequence of digits, each digit represented by the equivalent ASCII value (e.g. digit 7 is represented by 37 hex). ASCII B (42 hex) can also be incorporated anywhere within the sequence to introduce a 4 second delay before sending the next digit(s).